Weekly Updates (2/11/2023) – Krishna Tarun Saikonda

This report details the procedure of transforming a traffic network initially in the Simulation of Urban Mobility (SUMO) XML format into OpenDrive, subsequently into Lanelet2, and ultimately visualizing it using Common road and JSOM (Java OpenStreetMap Editor). This conversion process aims to enable the utilization of traffic network data across various simulation and design platforms, thereby improving the potential for testing and refining algorithms for autonomous vehicle development.

The report further delves into the intricate process of categorizing vehicles based on their types using Floating Car Data (FCD) alongside additional pertinent datasets. It meticulously explores and applies various machine learning models to distinguish and classify these vehicle types. Through an in-depth analysis, the report evaluates the efficacy and accuracy of these diverse machine learning models in their performance for such classification tasks.

Lanelet2 Conversion:

Lanelet2 is a specialized data format used to represent the intricate details of road networks. Its necessity stems from its ability to provide a highly detailed and accurate representation of road infrastructure, including lane markings, traffic rules, and other essential traffic-related information. This format is crucial in various fields, especially in the development and testing of autonomous vehicles, as it allows for precise modeling of road structures and traffic regulations. Lanelet2 facilitates the creation of highly detailed maps, essential for simulating and analyzing complex traffic scenarios, enabling better testing and validation of autonomous driving algorithms. Its use enhances the development and testing of self-driving technology by offering a standardized, detailed, and machine-readable representation of road networks.

Methodology:

Converting a SUMO .net file to Lanelet2 format using Command Road Designer involves a series of steps to ensure a successful transformation of the traffic network data.

Understanding the Tools:

SUMO .net file: This file contains network topology information used in traffic simulations and represents the road network in the Simulation of Urban Mobility (SUMO) format.

Lanelet2: This format provides detailed representation for road networks, including lane markings, traffic rules, and other essential traffic-related data.

Command Road Designer: A tool for creating, editing, and visualizing scenarios for autonomous driving simulations, compatible with various map formats, including Open Street Map (OSM)

The initial stage involves preparing for the conversion process by acquiring access to the Command Road Designer tool, typically used for converting SUMO .net files to Lanelet2 format. This necessitates obtaining the specific SUMO .net file containing the essential traffic network data. Upon opening Command Road Designer, the appropriate function or command responsible for the conversion is identified, whether through a command-line interface or a graphical user interface (GUI). Executing the conversion involves inputting the necessary parameters, such as the file path of the SUMO .net file and relevant settings, prompting the Command Road Designer to interpret the network topology and convert it into Lanelet2 format.

Lanelet2 Format:

The OpenStreetMap (OSM) format used in Lanelet2 will have a specific structure reflecting detailed road network information. In the .osm file, Lanelet2 utilizes a specialized representation that includes various elements to describe road features:

Nodes:

Nodes represent individual points on the map, typically defining coordinates (latitude and longitude) of specific locations or junctions on the road.

```
<node id=1 lat='43.50934231789' lon='-80.53670525588' />
```

Ways:

Ways are ordered lists of nodes representing linear features such as roads, lanes, or other linear elements. They define the shape and geometry of roads and lanes.

```
<way id='10'>
    <nd ref='2' />
    <nd ref='3' />
    <nd ref='4' />
</way>
```

Relations:

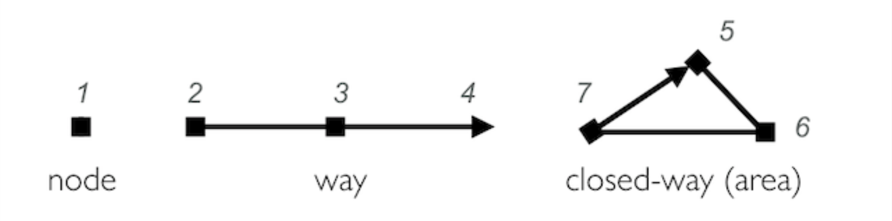
Relations organize different elements together, allowing for the representation of complex structures like intersections, traffic lights, or other related features. They establish connections between various nodes and ways, defining their functional relationships.

Tags:

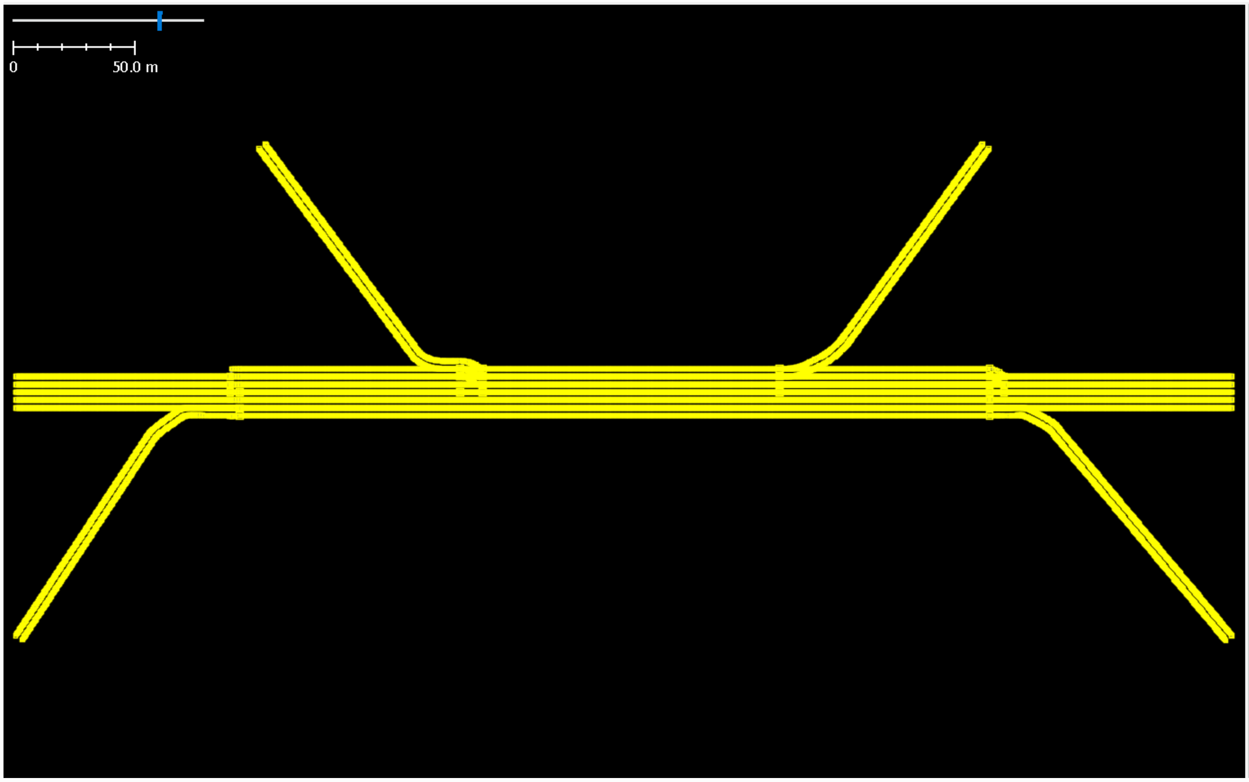
Tags are key-value pairs assigned to nodes, ways, and relations to provide additional descriptive information about the features. These tags include information such as road types, lane markings, traffic rules, speed limits, or other attributes that define the road network's characteristics.

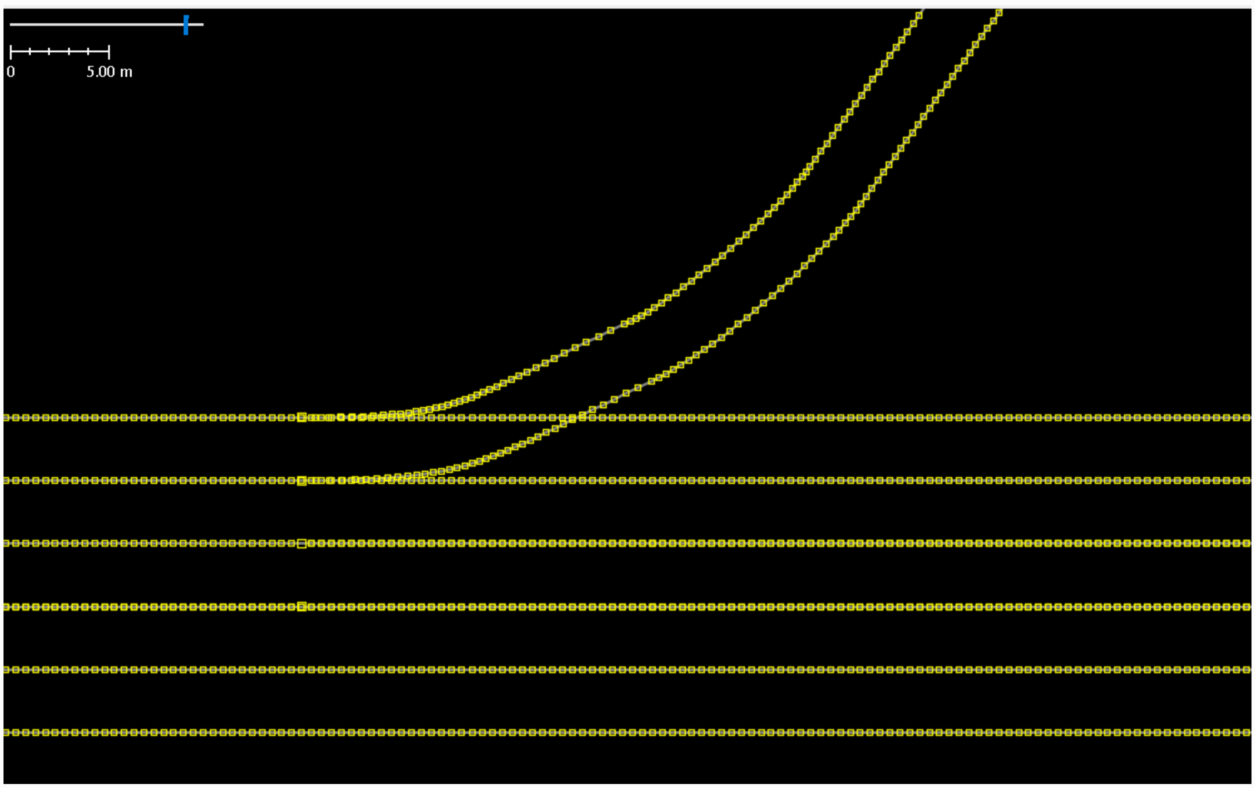
```
<node id='-1' lat='43.5094' lon='-80.5367'>
   <tag k='gs' v='vehicle'/>
   <tag k='name' v='leading_vehicle'/>
</node>
```

Example:



Representation of lanelet2 map:





SCENARIO 2 - Exploring potential predictions or classification models to determine vehicle types under specific configurations using aggregating the columns of FCD Output using mean,standard deviation and variance.

To explore predicting or classifying vehicle types based on specific configurations using Floating Car Data (FCD) output, I aggregated the columns by calculating statistical measures such as the mean, standard deviation, and variance. This process involves summarizing the FCD data attributes, grouped by various configurations, to potentially create a basis for predictive modeling. By calculating these statistical measures for each column, i aim to discern patterns or characteristics specific to different vehicle types.

Features Used:

Time: This represents the time in seconds when the data was recorded. It indicates the moment at which the observations were made.

ID: This is an identifier associated with each data entry. It likely serves as a unique identifier for different observations.

x and y: These columns represent the X and Y coordinates of the vehicle's location. These coordinates indicate the position of the vehicle on a map, with X and Y denoting the horizontal and vertical dimensions.

Angle: This feature denotes the angle at which the vehicle is oriented. It specifies the direction the vehicle is facing in degrees.

Type: This indicates the type or category of the vehicle, such as "SportsCarDriver," "CautiousDriver," "ElderlyDriver," or "AggressiveLanechangingdriver."

Speed: This represents the vehicle's speed in units per second (e.g., meters per second or kilometers per hour), indicating how fast the vehicle is moving.

Pos: This feature likely refers to the vehicle's position or location, possibly providing additional context to its position within the dataset.

Acceleration: It represents the acceleration of the vehicle, which is the rate of change of its speed over time, typically measured in units per second squared.

CO: This column might indicate the vehicle's carbon monoxide emissions or another relevant environmental factor.

Fuel: This could be the amount of fuel consumed by the vehicle or some other related metric.

Noise: This feature might represent the noise level generated by the vehicle, which can be relevant for environmental or noise pollution considerations.

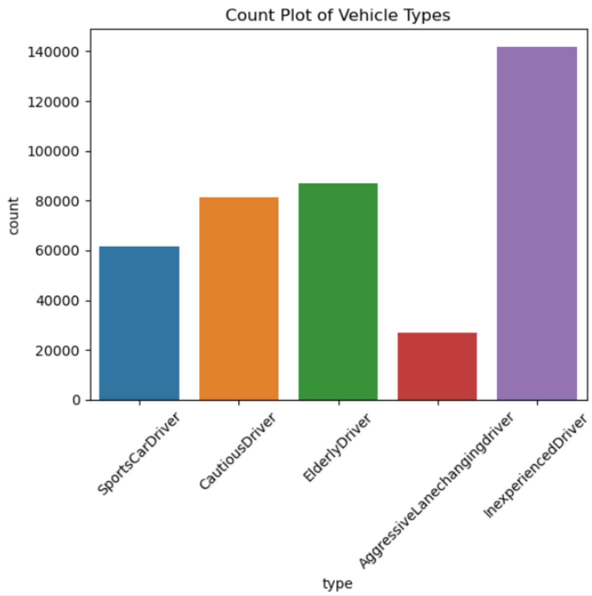
Lane_length: This could be the length of the lane that the vehicle is in or related to the road infrastructure.

Lane_change: It appears to be an indicator of whether the vehicle is changing lanes, with "0" indicating no lane change and "1" indicating a lane change.

Vehicle_density: This feature may indicate the density of vehicles in the vicinity, providing information about traffic conditions.

Avg_speed_nearby_vehicles: This feature likely represents the average speed of vehicles in the vicinity of the observed vehicle.

Vehicle Type Count Plot:



Classification Models:

Random Forest is a machine learning approach that involves creating numerous decision trees during training and determining the class that most trees agree upon. This method is prized for its reliability and precision, drawing on the predictions from various trees that individually capture different facets of the data.

Deep Neural Networks (DNN) are sophisticated neural networks with multiple hidden layers, adept at uncovering intricate patterns within datasets. They excel in capturing complex relationships but necessitate extensive data and computational resources for efficient training.

K-Nearest Neighbors (KNN) is a straightforward algorithm that stores all available cases and categorizes new instances based on similarity measures, typically employing distance functions. KNN has demonstrated success across various classification and regression scenarios, including those involving nonlinear relationships.

Gaussian Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem and the assumption of independence among predictors. It calculates the probability of a given instance belonging to a particular class by assuming that the features are conditionally independent. Despite its simplicity and the assumption of feature independence, Gaussian Naive Bayes is known for its effectiveness in various classification tasks.

Results:

Model	Accuracy
Random Forest	0.7451
k-NN	0.45
DNN	0.4902

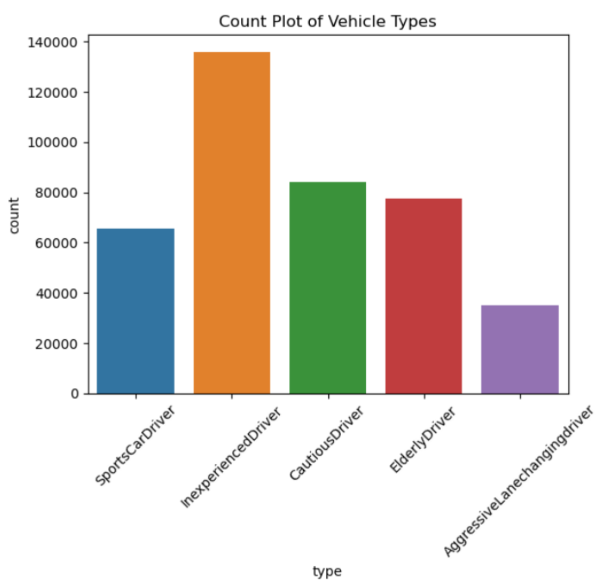
Model	Accuracy
Gaussian NB	0.6039

High accuracy for the Random Forest model may be attributed to its ensemble learning technique, combining predictions from numerous decision trees. This method is known for its robustness and effectiveness in handling complex datasets. The individual trees within the Random Forest capture different aspects and nuances present in the data, which collectively contribute to a more accurate overall prediction. Each tree in the forest operates independently, considering a subset of the data and features, which helps in reducing overfitting and bias that might be present in a single decision tree model.

Model performance on unseen data:

After achieving the highest accuracy with the Random Forest model trained on a specific dataset, I proceeded to apply this well-performing model to a novel test dataset. This new test dataset was derived from a recent simulation carried out within SUMO, representing fresh and previously unencountered data. By utilizing the Random Forest model trained on the original dataset, I sought to evaluate its predictive capabilities and generalizability on this new and distinct test data. This process aimed to determine the model's effectiveness in making accurate predictions when confronted with previously unseen simulated scenarios, further assessing its robustness and reliability beyond the initial training data.

Vehicle Type Count Plot of New data:



Results:

Model	Accuracy
Random Forest	0.7130503144654088

SCENARIO 3 - Exploring potential predictions or classification models to determine vehicle types under specific configurations using the FCD Output.

To explore predicting or classifying vehicle types based on specific configurations using Floating Car Data (FCD) output. This process involves summarizing the FCD data attributes, grouped by various configurations, to potentially create a basis for predictive modeling. By calculating these statistical measures for each column, i aim to discern patterns or characteristics specific to different vehicle types.

Features Used:

The dataset incorporates the following key features for each entry: time, id, x and y coordinates, angle, driver type, speed, position (pos), lane, slope, and acceleration. These attributes provide detailed insights into the spatial positioning, orientation, driver type, velocity, lane, and other relevant parameters of the observed vehicles over time.

Classification Models:

I used the same classification model as the above which are Random Forest, Deep Neural Networks (DNN),K-Nearest Neighbors (KNN),Gaussian Naive Bayes

Results:

Model	Accuracy
Random Forest	0.9597470387472395
k-NN	0.86
DNN	0.4480
Gaussian NB	0.41406093153985146

Model performance on unseen data:

After achieving the highest accuracy with the Random Forest model trained on a specific dataset, I proceeded to apply this well-performing model to a novel test dataset. This new test dataset was derived from a recent simulation carried out within SUMO, representing fresh and previously unencountered data. By utilizing the Random Forest model trained on the original dataset, I sought to evaluate its predictive capabilities and generalizability on this new and distinct test data. This process aimed to determine the model's effectiveness in making accurate predictions when confronted with previously unseen simulated scenarios, further assessing its robustness and reliability beyond the initial training data.

Results:

Model	Accuracy
Random Forest	0.5632560053804997

SCENARIO 4 - Exploring potential predictions or classification models to determine vehicle types under specific configurations using the FCD Output with other relevant feature extracted from other output data.

In Scenario 4, the objective is to explore and develop prediction or classification models to identify vehicle types within configurations. This involves utilizing the Floating Car Data (FCD) Output alongside other relevant features extracted from additional output data sources. The intention is to combine data from the FCD output with supplementary information extracted from other related sources to enhance the accuracy and reliability of predicting or classifying different vehicle types within specific scenarios or configurations. The goal is to create models that leverage a comprehensive set of features from various data outputs, aiming to achieve improved accuracy in discerning and categorizing diverse vehicle types operating within distinct configurations.

Features Used:

The dataset incorporates the following key features for each entry: time, id, x and y coordinates, angle, driver type, speed, position (pos), lane, slope, and acceleration. These attributes provide detailed insights into the spatial positioning, orientation, driver type, velocity, lane, and other relevant parameters of the observed vehicles over time.

Lane_Change: This feature likely indicates whether the vehicle is changing lanes. A value of "0" may denote no lane change, while "1" could indicate that the vehicle is currently in the process of changing lanes.

Vehicle_Density: This feature represents the density of vehicles in the vicinity of the observed vehicle. It provides information about the level of traffic in the area.

Avg_Speed_Nearby_Vehicles: This feature likely denotes the average speed of vehicles that are in close proximity to the observed vehicle. It provides insight into the speed of neighboring vehicles.

Classification Models:

I used the same classification model as the above which are Random Forest, Deep Neural Networks (DNN),K-Nearest Neighbors (KNN),Gaussian Naive Bayes

Results:

Model	Accuracy
Random Forest	0.9686182493475206
k-NN	0.90
DNN	0.4462
Gaussian NB	0.4064821321019875

Model performance on unseen data:

After achieving the highest accuracy with the Random Forest model trained on a specific dataset, I proceeded to apply this well-performing model to a novel test dataset. This new test dataset was derived from a recent simulation carried out within SUMO, representing fresh and previously unencountered data. By utilizing the Random Forest model trained on the original dataset, I sought to evaluate its predictive capabilities and generalizability on this new and distinct test data. This process aimed to determine the model's effectiveness in making accurate predictions when confronted with previously unseen simulated scenarios, further assessing its robustness and reliability beyond the initial training data.

Results:

Model	Accuracy
Random Forest	0.5756833535771289

Analysis:

- In Scenario 2, aggregating FCD features resulted in moderate accuracy for the Random Forest model, outperforming other models significantly.
- In Scenario 3, utilizing only FCD output for classification resulted in exceptional accuracy for the Random Forest model, significantly higher than other models.
- Scenario 4, which involved incorporating additional relevant features along with FCD data, yielded the highest accuracy for the Random Forest model, indicating the value of supplementary information in enhancing classification accuracy.
- Across all scenarios, Random Forest consistently demonstrated the highest or competitive accuracy, highlighting its efficacy in vehicle type classification, especially when supplemented with additional relevant features beyond raw FCD output.

Analysis on unseen data:

The provided scenarios indicate the performance of a Random Forest model in different contexts of vehicle type classification based on specific configurations using varying sets of features and data.

1. Scenario 2:
   - Approach: Aggregating columns of the Floating Car Data (FCD) Output using mean, standard deviation, and variance.
   - Model Performance: The Random Forest achieved an accuracy of 0.7131 on unseen data. This method seems to leverage statistical aggregation, potentially capturing essential patterns across varied vehicle types and configurations.

2. Scenario 3:
   - Approach: Utilizing only the FCD Output for classification.
   - Model Performance: The Random Forest model produced an accuracy of 0.5633 on unseen data. This lower accuracy might imply that using only the raw FCD data may not capture sufficiently nuanced features or configurations for accurate classification.

3. Scenario 4:
   - Approach: Exploring vehicle type predictions using the FCD Output along with additional relevant features extracted from other output data sources.
   - Model Performance: The Random Forest model achieved an accuracy of 0.5757 on unseen data. Incorporating supplementary data seemingly provided a slight performance improvement, suggesting that additional relevant features may contribute to enhanced classification accuracy.

Analysis:
- Scenario 2 demonstrates that statistical aggregation of FCD features, like mean, standard deviation, and variance, results in the highest accuracy. This method likely captures more comprehensive and discerning patterns in the data.
- Scenarios 3 and 4 indicate that solely relying on the FCD data for classification might not be as effective as incorporating other relevant extracted features from additional data sources. In both cases, the accuracy was marginally higher when additional features were included, implying a potential benefit in using a broader set of relevant data for more accurate classification.

These outcomes suggest that incorporating more diverse and relevant features beyond raw FCD data could enhance the accuracy of vehicle type classification models, as observed from the varying accuracies across these different scenarios.